下载APP

05 | 流程把控：控好流程，让面试进程高效有温度

2021-04-15 四火

技术面试官识人手册

进入课程 >

讲述：四火

时长 17:30　大小 16.03M

你好，我是四火。

“工欲善其事，必先利其器”，在专栏前面的部分，我们已经讲了怎样理解团队的需要，怎样做面试计划，以及怎样设计合适的技术问题，这些，可以说都是在正式面试之前需要搞定的事情。

今天这节课，我将会和你分享面试流程把控的基本思路，以及一些常用技巧。只有把控好流程，我们前面辛苦设计的面试计划、技术问题才能真正派上用场。

流程把控的反面例子

先来问一个开门见山的问题，究竟谁来把控面试流程，候选人还是面试官？

有人说是候选人，当候选人带着自由度表现的时候，才能发挥自己的实力；有人说是面试官，因为面试官需要按照预想，领着候选人来推进面试进展。

好像都有道理对不对？我卖个关子，先不回答这个问题，带着这个问题，我们来看看几个流程把控的反面例子。看看这些例子中，有没有你熟悉的。

收尾时间狂飙代码

这可能是一个一下就映入脑海的反面例子了，也是我见到的最多的一种流程把控的反面例子。

无论是由于问题过于复杂、沟通出现误会、候选人解决问题能力不强，或是候选人或面试官迟到等等原因，**到了白板编码环节，候选人有了思路，也开始写了，可时间快没了。**

这时候直接放弃吧，有点可惜，于是候选人就擦擦额头上的汗珠，颤抖地拿起记号笔，开始马力十足地在白板上狂奔，字也是越写越龙飞凤舞；而面试官呢，也颇为体谅，让下一位面试官在门外稍加等待，要求延期“就五分钟”，并不断提醒候选人时间越来越少。

最后，无论代码是否写完，面试匆匆结束，就仿佛一场大战结束，候选人累得瘫坐下来，这一轮的面试官一只脚刚迈出会议室，下一轮的面试官另一只脚就迈进来。若是代码基本写完了，候选人心里想，我去，真够紧张刺激，幸亏老子搞定了；若是没写完，可得失落和恍惚好一阵子。结束的时候，双方礼貌性地握手，手心满是汗。

寒暄时间占比不当

这里的“寒暄时间”，指的是技术面试中，正儿八经讨论技术问题以外的时间，一般发生在面试开始时和结束前。

一种极端情况是这个时间过短，甚至没有。候选人和面试官见了面，握个手，就开始做题，连个过渡都没有，让人觉得颇为唐突——“是不是这个团队都是老古板啊？”。

另一种极端情况是，打个招呼，互相介绍一下，聊两句感兴趣的话题，或是最近的热点新闻。扯扯淡，聊聊天，帮助候选人放松一下。本来这些都是挺好的“操作”，但问题是这个时间如果拖得太长，原本的技术考察重点就打了折扣。

我分享一个我 shadow（观摩）的经历，有一位面试官在面试中和候选人“闲聊”，面试的一半时间过去了，候选人终于忍不住了，她主动问面试官，“咱们不谈点什么技术问题吗？还是继续这样随便聊吗？”，他们这才开始讨论技术。

考虑到在面试计划的时候，这位面试官是确定要考察算法和数据结构的，因此事后我问面试官，我理解这也是一种面试风格，可既然面试计划已经确定了，你为什么没有主导这个面试过程，让技术方面的讨论早一点开始？面试官回答说，他想看看候选人有没有足够的主动性主导这个面试过程。

对于这个案例，我的想法是，面试官这样做，虽然考察了他所关心的“候选人的主动性”这个数据点，但也使得这个面试偏离了原本的计划，也就是说，在我看来，确有所得，但得不偿失。

踌躇不定的思路选择

这个例子是来自于我自己。

一个问题往往有很多解决的思路，有一次，候选人和我讨论某个问题的算法，这个过程一开始还挺顺利，候选人选择使用 Map+ 数组排序的思路做，方案虽然略有一些硬伤，但是也行得通；期间他忽然想到，使用一个二叉搜索树数据结构的思路更好，于是我们又放弃了前面的思路，沿着这条路推进；结果忙活到一半，他忽然再一次“恍然大悟”，觉得可以用一个 Map+ 链表的方式来做更简单。

这整个过程其实没有太大的问题，整个思路的转变，反映了候选人很好的思考能力，能够跳出具体实现，重新审视问题。

但是由于这一系列的思路调整耽误了太长时间，导致到面试结束的时候，我们只讨论了思路，连完整的方案都没有讨论清楚，自然原本计划的编码根本没有时间开展。从考察角度来说的话，我们就丢失了相当一部分考察项的覆盖。

流程把控的技巧

不知道刚才那几个例子是否给你带来了似曾相识的感觉？其实这些反例说到底还是因为流程把控没做好。

那既然如此，下面我就来正面谈一谈我的理解，带你一起看看有哪些流程把控的技巧。这些技巧能够帮助我们在面试中，**保证最核心的考察项能够覆盖到，同时，还能够给候选人留下一个好印象。**毕竟，如同我们在专栏前文中所说，面试是“双向”的。

有头有尾，有礼有节

首先我要谈的，不是什么复杂的技巧，而是礼节。

什么，礼节？对，你没有看错。我认为，面试官有义务确保考察过程的完整性，这能让候选人心情愉悦，在友好的氛围里完成面试的过程。

不知你是否听说过，有一些面试的风格颇为激进，甚至不尊重人，观察候选人在极端压力下的表现。也许这确实可以让面试官得到他想要的考察数据，但我认为这是非常不妥的。

面试，说到底是社交活动的一种，我们必须建立在平等和尊重的基础上，我们要达到目的，但不能不择手段，因此在我看来，这就是一个根本性的错误。

我觉得面试就有点像是江湖上相逢，缘分一场，我们客客气气地切磋武艺，点到为止，你我互相尊重，为的也是来日好再相见。

面试应该做到有头有尾，在面试开始时，我们可以：

- 礼貌地打招呼、握手，邀请候选人坐下，问问今天感觉怎么样，如果迟到了要表示歉意；
- 询问候选人是不是需要休息两分钟，是不是需要喝水，是不是需要使用洗手间，在不少互联网大厂的面试培训里，这一条都被反复提到；
- 做自我介绍，这个自我介绍可以很简单，一两句话概括，但是让彼此认识的环节不能少；
- 可以简单提一提，在今天的面试中可以预期的内容，这样候选人会有个心理准备。

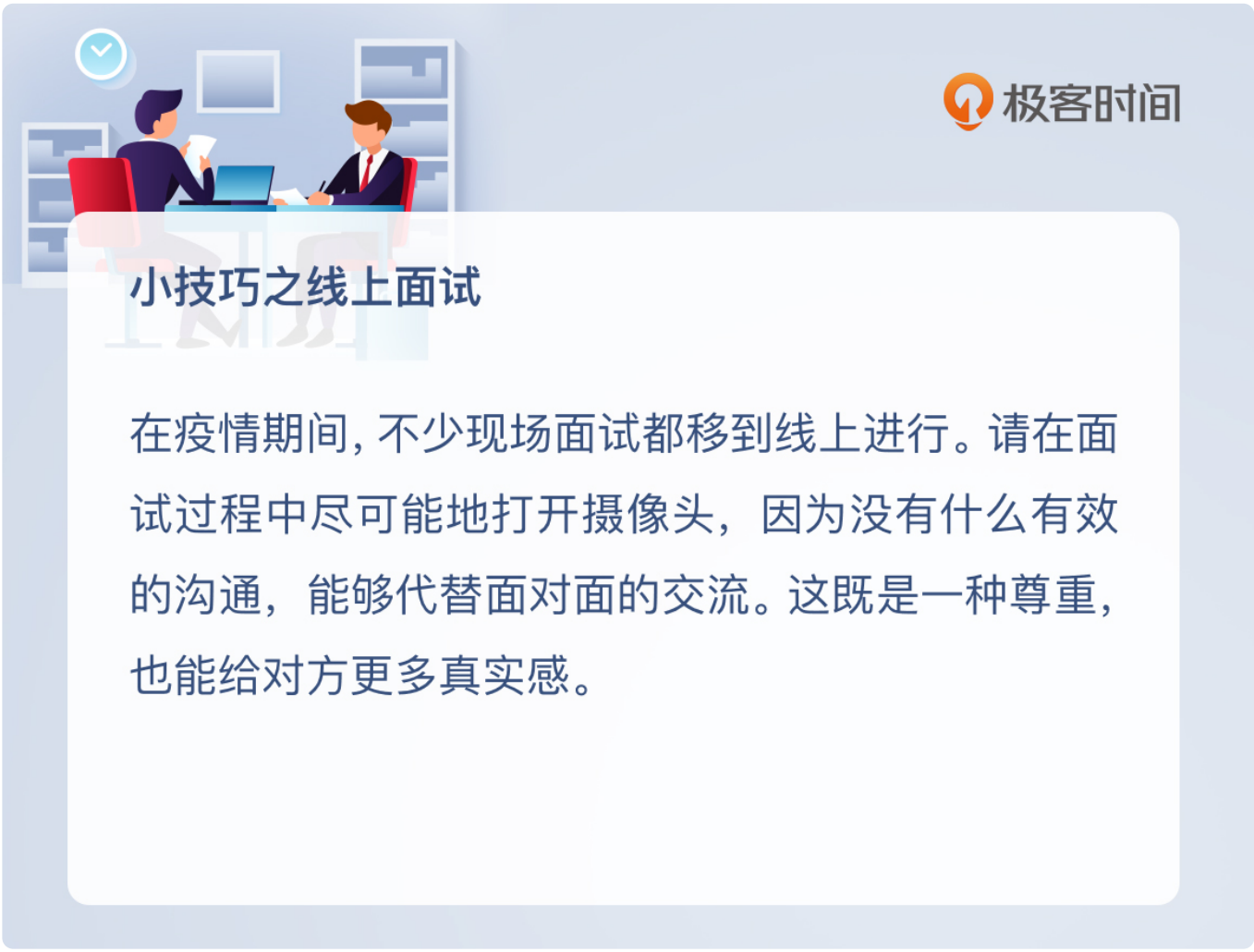
而在面试结束前，我们可以：

- 表示问题的讨论告一段落，留给对方问问题的机会，表示在自己力所能及的范围内将努力回答对方的问题；

对于候选人的到访表示感谢，可以赞许一下今天的讨论，也可以对他后续的求职过程表示祝愿；

礼貌地握手、告别。

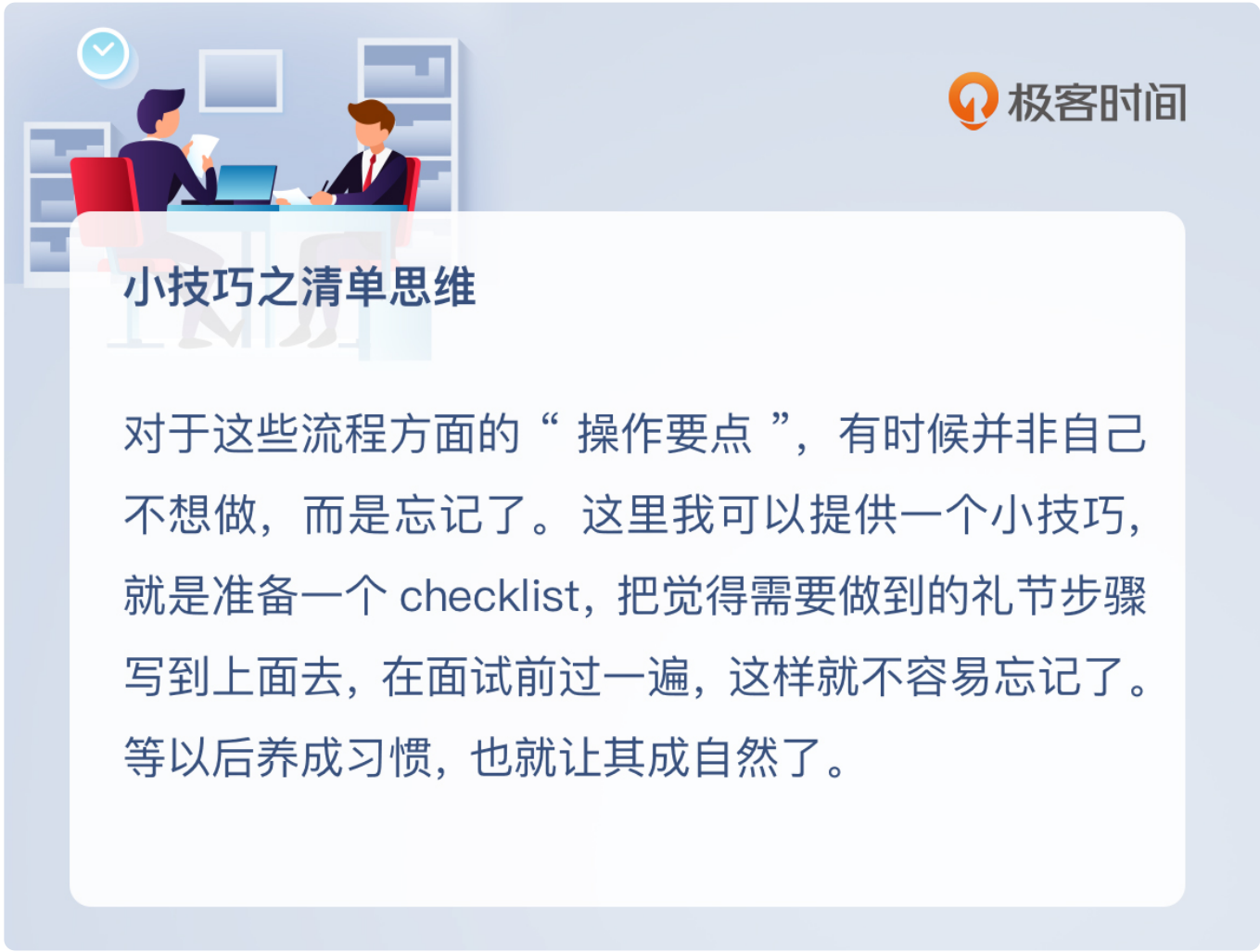
对于电话面试，或者是特殊时期的线上面试，不能握手了，可上面大部分礼节还是一样的。上述这样的过程也许用不了 5 分钟、10 分钟，但是面试的整个过程显得自然、得体，并且也能够帮助候选人放松。



你看，这些有什么特别的技巧吗？完全没有，**上面这些哪是在说面试啊，其实只是礼仪之邦的待客之道对不对？**

没错！其实，要做到上面这些，只需要细心，并不需要多么高深的技巧，但是却能够令候选人大大提升印象分。还是那句话，面试是双向的。面对多家竞争对手的时候，你肯定希望优秀的候选人能够选择你的公司和团队。

我说一个我自己的故事，那是在好几年前了，我在找工作期间，在国内几个城市都面试了几家公司，只有当时在北京的亚马逊，面试的过程中，面试官主动给我倒水，面试完毕后送我离开（当时需要上出租车，赶去机场）。且不说面试的内容，这些细节给了我非常正面的印象，我相信无论业务和技术怎样，面试我的团队至少是非常懂得尊重人的。



做一个完整的项目

对于技术面试而言，**面试官要抱着和候选人一起做一个有头有尾的迷你项目的心态。**

这里，“迷你项目”这四个字可以说是重中之重，即所谓“麻雀虽小而五脏俱全”。这个问题的解决过程要完整，从需求澄清，到分解、设计、抽象、落地、验证、改进，有头有尾。

举例来说，前文所谈到的这个网约车的问题，我们考察的是数据结构和算法，在代码层面“落地”的思路沟通清楚以后，就可以写白板代码了。

在代码完成后，候选人可以使用一个简单用例走读代码，验证正确性。再之后，面试官还可以提出“follow-up”问题，比如聊一聊有什么瓶颈，怎么改进。

好，我想从中你可以看到这样一个完整的过程，而**理想的面试，确实也不应该在中间的某一个过程戛然而止。**

比方说，如果代码实现的思路讨论清楚了，但是因为没有时间完成代码实现，想必候选人也会觉得沮丧；而只有一个完整的过程，才会让候选人如释重负，有一种“项目完成了”的成就感。

举例来说，一个模糊的问题，一个代码层面考察（例如算法和数据结构）的路径，一轮面试总时间为 1 小时，而去除寒暄和收尾的环节，核心时间为 50 分钟，那么可以参考这样一个时间分配的思路：

10 分钟澄清问题，厘清面试中需要实现的需求，细化到可讨论实现的层面；

15 分钟分析、讨论该需求抽象后的问题，这些问题必须是软件可解的，比如该使用怎样的数据结构和算法来解决，时间、空间复杂度各是多少；

15 分钟现场编码；

5 分钟使用实际用例走读代码，验证正确性；

5 分钟讨论算法的优化，以及进一步的改进空间。

你可能会问，现场编码只有 15 分钟，这么短的时间，够吗？

够。这里我必须说明的是，**在编码前，我们期待候选人和面试官已经将思路大体上沟通清楚了**，那么这个编码过程，其实只不过是一个将思路“翻译”成代码的过程而已。

我们在编码前对于思路和逻辑的讨论，目的就是要保证编码能够顺利进行，所有的关于“problem solving”的部分基本已经解决，于是就只剩编码过程。

值得注意的是，编码前的思路和讨论，以及编码本身，我们考察的侧重点是不一样的。比方说，有的候选人就是具备很强的逻辑能力，来解决问题，但是就是无法将思路落实到代码上。因此，别看过程很短暂，却是不可缺少的。

你可能也注意到，在需求确认以后，很多候选人都喜欢上来就立马落笔写代码，但作为面试官来说，是不是可以落笔，需要着重关注候选人是不是已经有了清晰的思路，否则很可能浪费时间。这样急于落笔也很难一蹴而就，来回涂改，会做很多无用功。

如果你留意到，候选人即便有了清晰的思路，在基础编码能力具备的情况下，代码的实现依然要花费超过 15 分钟，那么你就应该考虑调整这个问题的设计了，是不是要求代码实现的部分过于复杂，从而最终导致在限定时间内，整个问题的解决流程无法完成。

如果是这种情况，**我们可以要求只实现代码片段，例如算法的核心部分，或者是涉及到的几个重要的类，而不是整段代码。**通过这种方式，我们可以将预期的编码时长降下来。

举例来说，如果实现一个完整的 LRU 缓存颇为耗时，那么算法实现可以要求写出数据结构，以及其中缓存更新的部分。

另外，你可能还听说过，在有些面试中，面试官会在一轮面试中，要求实现两个较为简单的算法。这种情况确实有，但是我一般是不推荐的，原因就是，这整个问题的讨论和解决过程已经颇为耗时了，很难在一轮面试中覆盖一个以上的算法实现。

如果硬要塞进两个算法题的实现，那就要问非常简单的问题，那么，即便勉强做到了，用意又何在呢？

选择合适的介入时机

还记得这一讲一开始，我开门见山地问的那个问题吗？“究竟谁来把控面试流程，候选人还是面试官？”

我打个比方，一个国家的经济活动有强有弱，走势有高有低，通常情况下，从大体上看，市场经济可以自然地发展，但是在某些特殊环节，或者极端情况下，国家需要介入，需要调控。

面试也是如此。我认为，在核心时间中，面试官应该关注于大的层面上，是不是能够顺利完成今天的考察内容，是不是能够得到自己关心的考察数据。

如果候选人能力很强，对于问题的推进合理有效，那么面试官就要学会做“绿叶”，给对方更多的发挥空间，让对方去主导问题的解决，这当然是最理想的情况，候选人能跟着自

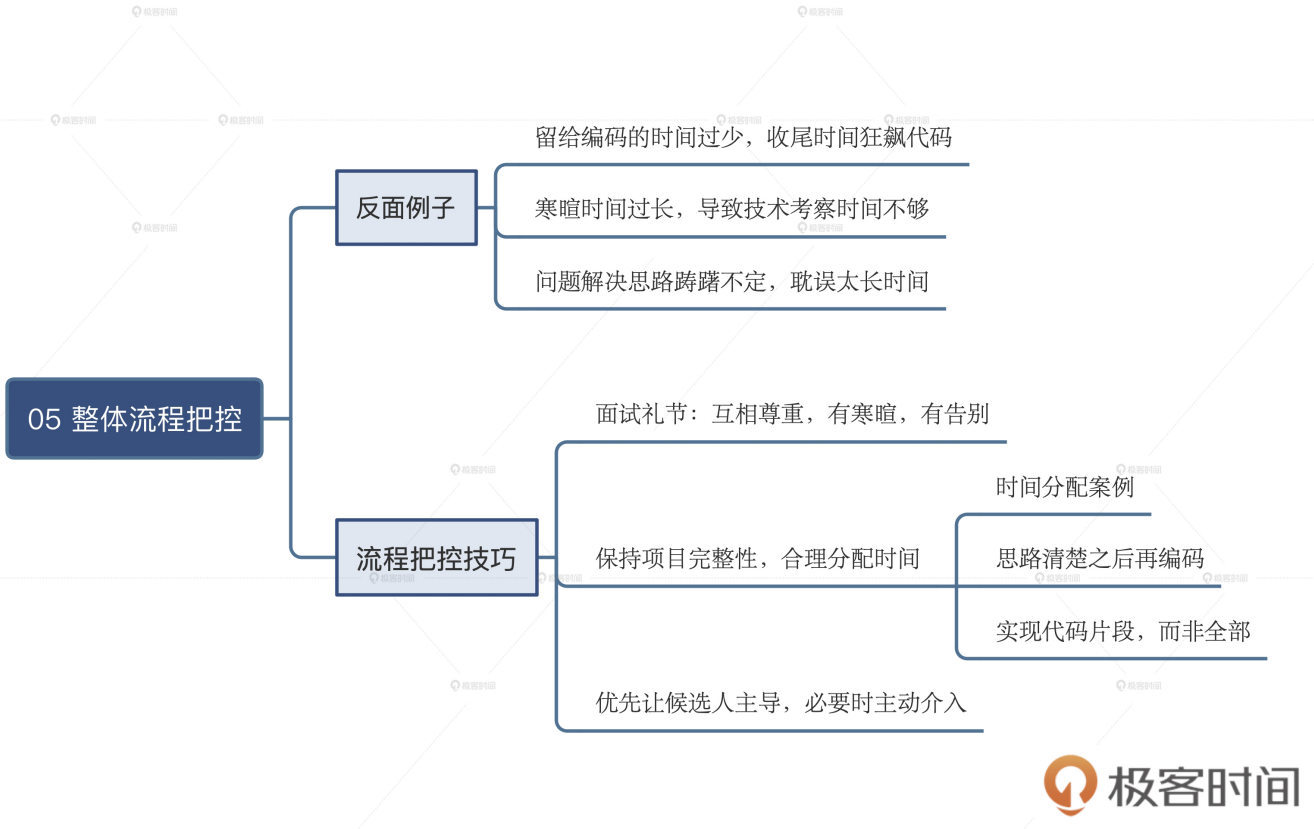
己的计划走，自然也最舒服。

反之，如果遇到了困难（还记得我们上一讲在“设定‘踮踮脚能够到’的最高难度”部分，谈到的降低难度的方法吗？），或者看着路线要“跑偏”，或者时间进度已经大大落后，那么面试官就要毫不犹豫地及时介入，主动调控，从而保证整个方向和时间的可控。

总结与思考

好，今天我们开始讲怎样主导技术面试，把控流程了，既谈到了一些常见的反面例子，又介绍了我们应该掌握怎样的技巧，完成一个成功的技术面试。

在这一讲的最后，我想再次强调“互相尊重”的重要性，虽然我不会在专栏中再反复强调它，但请记得它是我们面试等一切职业行为的最基本要求。



最后，我有一个问题，正在阅读的你，能否在评论区和大家一起聊一聊，你作为候选人参与过的面试，都有哪一些有意思的细节呢？期待你的分享。

好，我是四火，我们下一讲见！

提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 答疑课堂01：面试计划篇热点问题解答

下一篇 06 | 算法和数据结构考察：怎样有层次地驾驭算法考察？

精选留言 (3)

写留言

蓝冰
2021-04-26

我很认同要尽量保证面试的完整性，尽可能的给候选人提供一个舒适的环境，让候选充分发挥自己的才能，我们才有机会选择更合适的人选

展开

作者回复: 对，面试官其实是和候选人站在一边的，而不是对面。因为他们在这一点上的目的是一致的。

1

楼下小黑哥
2021-04-21

之前几次面试阿里，每次最后听到感谢你参加今天的面试、第二天就收到不合适的邮件😅

展开

作者回复: 其实啊，不说这个结果，这种操作过程还是挺好的，无论结果如何，都应该诚恳地感谢候选人来参加面试。而且第二天就收到邮件，可以说工作效率也很高。当然，有些公司能够电话亲自沟通，提供一部分反馈，那样更值得尊敬。“默拒”才是最糟糕的。

小风
2021-04-21

作为没有什么经验的面试官，不知道什么的问题是能够抓住重点的，小公司也没什么培训，有点被赶鸭子上架的感觉！

展开

作者回复: 没错，这个问题挺常见的，希望这个专栏在这一点上对你有用。